

SELECTED HEURISTIC ALGORITHMS FOR SOLVING JOB SHOP AND FLOW SHOP SCHEDULING PROBLEMS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT FOR THE
REQUIREMENT OF THE DEGREE OF

BACHELOR OF TECHNOLOGY
IN
MECHANICAL ENGINEERING
BY
MANOJ KUMAR DAS
ROLL: - 110ME0286

UNDER THE GUIDANCE OF
PROF. S. K. PATEL
Department of Mechanical Engineering



Department of Mechanical Engineering
National Institute of Technology
Rourkela – 769008
2013-14



NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA – 769008

CERTIFICATE

This is to certify that the thesis entitled, “**SELECTED HEURISTICS ALGORITHMS FOR SOLVING JOB SHOP AND FLOW SHOP SCHEDULING PROBLEMS**”, submitted by Mr. Manoj Kumar Das in partial fulfillment of the requirement for the award of **Bachelor of Technology** degree in **Mechanical Engineering** at National Institute of Technology, Rourkela is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any degree/diploma.

Date:

Prof. S. K. Patel
Mechanical Engineering
National Institute of Technology
Rourkela-769008

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my project guide **Dr. S. K. Patel** for his exemplary guidance, monitoring and constant encouragement throughout the course of the project. The blessing, help and guidance given by him time to time will carry us a long journey of life on which we are about to embark.

I also express a deep sense of gratitude to all the faculty members and staffs of Mechanical Engineering Department, for providing me support and advice in preparing my thesis work and their valuable information and guidance, which helped us in completing the task through various stages.

Last, but not the least I would like to thank my parents and my friends who are involved directly and indirectly in successful completion of the present work.

Manoj Kumar Das

Roll: 110ME0286

ABSTRACT

Importance of job shop and flow shop scheduling has increased to a high extent. Nowadays, each and every industry focuses largely on how to schedule their machine working, since it is an important factor which decides the net productivity. All manufacturing systems including flexible manufacturing system follow a planned schedule of machine operation depending on the demand criterion. With the increase in number of machines and jobs to be scheduled, complexity of the problem increases which demands the need of a proper scheduling technique. Here this thesis shows some of the essential methods of solving a job shop and flow shop scheduling problems. This thesis focuses on finding the most efficient way of scheduling in a flow shop environment with the help of heuristics algorithms that include Palmer's algorithm, CDS algorithm and NEH algorithm. Comparison was also done between the various heuristics algorithms. For getting the optimum make span for job shop scheduling we have used branch and bound algorithm and shifting bottleneck algorithm. Basis input parameters are given in the problem which are then used for computing the make span. A C programming code was generated to find the optimum results of the scheduling problem.

LIST OF FIGURES

Fig.1: Information flow diagram in a manufacturing system

Fig.2: Results of branch and bound algorithm

Fig.3: Result of first iteration in shifting bottleneck heuristics

Fig.4: Result of second iteration in shifting bottleneck heuristics

Fig.5: Program output of Kusiak's algorithm

Fig.6: Program output of Johnson's algorithm

Fig.7: Gantt chart for shifting bottleneck heuristics

CONTENTS

<i>Topics</i>	<i>Page</i>
Chapter 1 Introduction	
1.1 Introduction	7
1.2 Classification Of Scheduling	9
1.3 Parameters Of A Scheduling Problem	10
1.4 Objectives	10
Chapter 2 Literature Survey	
2.1 Introduction	11
2.2 Literatures Review	11
Chapter 3 Methodology	
3.1 Introduction	15
3.2 Johnson's Algorithm	15
3.3 Kusiak's Algorithm	17
3.4 Branch and Bound Method	18
3.5 Shifting Bottleneck Algorithm	19
3.6 Heuristics Algorithm	
3.6.1 Palmer's Algorithm	23
3.6.2 CDS Algorithm	24
3.6.3 NEH Algorithm	26
Chapter 4 Results	29
Chapter 5 Conclusion	33
References	34

CHAPTER 1: INTRODUCTION

1.1 Introduction:

Scheduling is an important decision making process that is used on a regular basis in various manufacturing and service industries. It deals with the allocation of resources to tasks over given time periods and its goal is to optimize one or more objectives. The resources and tasks in an organization can take many different forms. The resources may be machines in a workshop, runways at an airport, crew at a construction site, processing units in a computing environment, and so on. The tasks may be operations in a production process, take-offs and landings at an airport, stages in a construction project, execution of computer programs, and so on. Each task may have certain priority level, an earliest possible starting time and a due date. The objectives can also take many different forms. One objective may be minimization of tasks completed after their respective due dates.

Scheduling, as a decision making process, plays an important role in most manufacturing and production systems as well as in most information processing environments. It is also important in transportation and distribution settings and in other types of service industries.

Example 1: A Roller Bearing Industry

Consider an industry that manufactures roller bearings for various uses in machine tool, spindle, automobiles and many more moving elements. The main raw material for such an operation are cylindrical rods. The production process consists of three stages: the printing of the logo, manufacturing balls from cylindrical rod and assembly of the roller balls with its component parts. Each stage consists of a number of machines which are not necessarily identical. The machines at a stage may differ slightly in the speed they operate, the life time till they operate or the maximum load they can handle. Each production order indicates a given quantity of a specific roller bearings that has to be

shipped by a committed shipping date or due date. In this case, our one objective of the scheduling system is to minimize the completion time so that job can be easily completed on time.

Fig.1 shows a brief overview of the scheduling environment is as follows: -

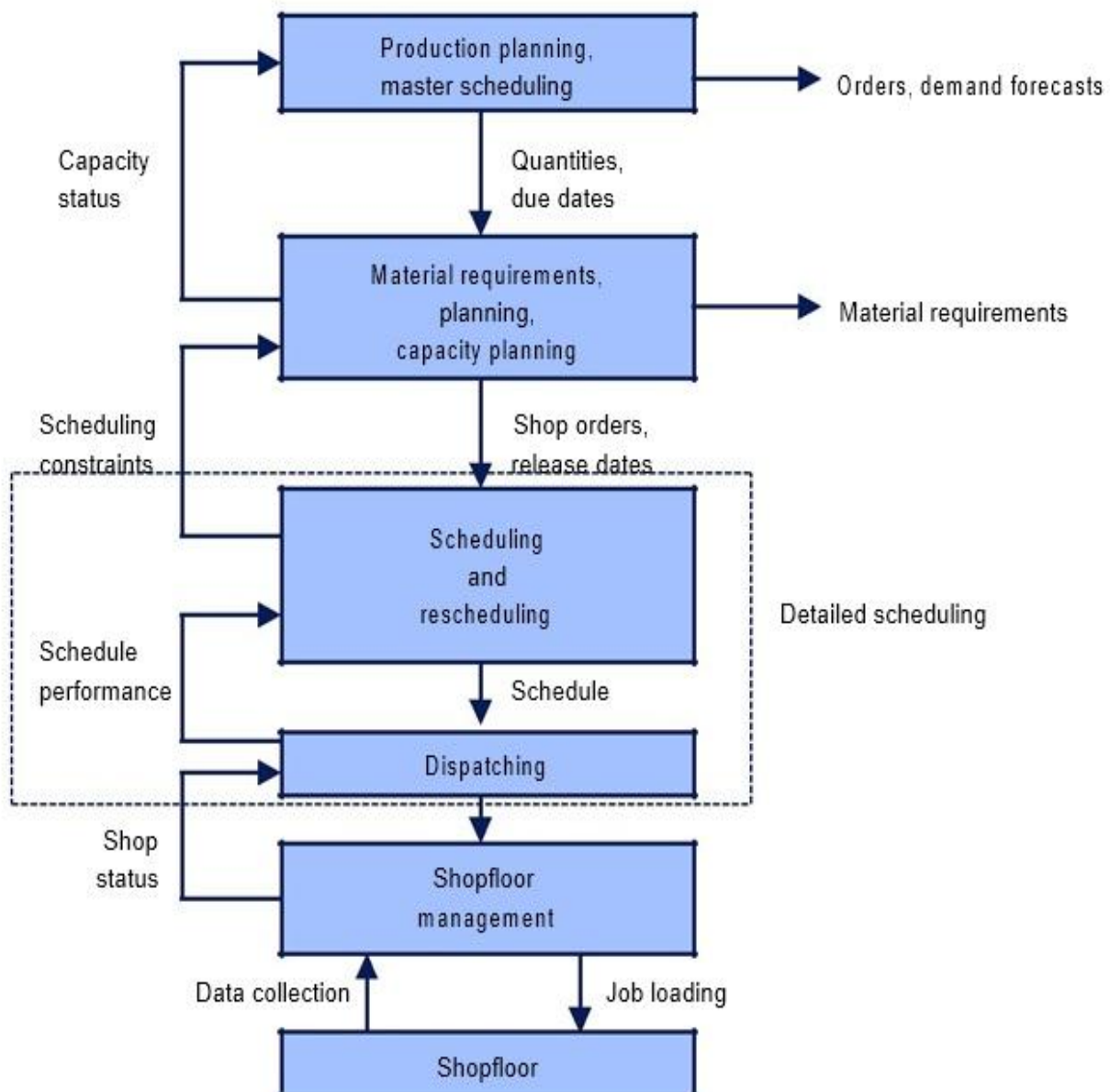


Fig.1: Information flow diagram in a manufacturing system [1]

1.2 Classification of Scheduling:

We can classify the scheduling problems basically into three types:-

a) Single Machine Scheduling:

In this type of scheduling jobs are first arranged in a particular order in a given machine. Our objective of minimizing the make span is best fulfilled when the arrangement of jobs is done on the basis of in process time, that means jobs with less in process time are put forward than those having higher in process times.

b) Flow Shop Scheduling:

The basic principle of this type of scheduling is that each machine takes the jobs in a particular sequence, which means each job has to get processed in each and every machine available in the flow shop environment. Also in each machine the in process time for each and every job is distinct. Therefore while evaluating a flow shop problem, we carry out different possible sequences of carrying out the job, and then the best among those is chosen.

c) Job Shop Scheduling:

The basic difference between job shop scheduling and flow shop scheduling is that unlike flow shop, here it is not compulsory for all the jobs to get processed in each and every machines available, that means each distinct job may be processed in any distinct number of machines as requirement demands. The sequence of jobs to be processed is also distinct in each machine. Based on the requirement of the problem, here also we choose the best sequence out of a number of possible sequences.

1.3 Parameters of a Scheduling Problem:

Based on the following parameters we can achieve our objective of optimally scheduling a task:-

- a) Total completion time
- b) Make span
- c) Lateness
- d) Total flow time
- e) Earliness
- f) Number of jobs corresponding to number of machines
- g) Due dates

1.4 Objectives:

This thesis is purely dedicated to the scheduling of jobs in flow shop and job shop environment, based on the following objectives: -

- a) To obtain a minimized make span value by different methods in job shop and flow shop scheduling problems.
- b) To obtain an optimal sequence for every machine and job. This would indicate the sequence of processing the jobs which would lead to achieve the desired result.
- c) To measure the performance of selected heuristics by comparing the results of each other in a flow shop environment.

CHAPTER 2: LITERATURE SURVEY

2.1 INTRODUCTION:

Scheduling is a very essential requirement for a stability and growth of the industry. Each order consists of varying objectives i.e. some may have objective of minimizing the make span, others may have to fulfill the demand within a particular due date failing in which may lead to penalties, few others may have the objective of minimizing the number of tardy jobs in the lot etc. Therefore the scheduling section does a lot of research on how to meet the customer demand in best possible way. Many researchers have carried out many different paths to achieve the same. They take into account the various parameters included and then find out ways to meet the demand. After doing a keen study on the various works carried out in this field, we have come to an understanding that a process which is best appropriate for one type of problem, is not suitable for the same problem with different objective. For example, if we try to minimize the make span we will end up in increasing the mean flow time and if we try to reduce mean tardiness it can lead to an increase in the mean flow time.

2.2 LITERATURE REVIEW:

Johnson [2] first presented an algorithm that can find optimum sequence for an n-job and 2-machine problem. Palmer [3] used a single iteration method to reach to the required result of minimizing the make span. Here the sequencing of the jobs were done based on the calculation of the slope index which are then sorted in the decreasing order so as to obtain the optimum sequence by palmer's method. This can be applied to m-machine and n-job flow shop scheduling problems.

The simple Johnson's algorithm was extended latter by Campbell et al [4]. Unlike the modified Johnson's algorithms it uses a number of iterations before reaching the final result. This method is a heuristics which is widely used and is commonly known as

Campbell, Dudek and Smith heuristics (CDS). Here a number of different sequences are proposed among which the optimal sequence is the one having the least make span. And at each and every iteration Johnson's rule followed.

Gupta [5] suggested a distinct algorithm for the procedure of minimizing the completion time and the make span of the sequence. A different technique is put up by him in obtaining the slope index, based on which the sequencing of jobs is done in the flow shop environment. Nawaz et al. [6] developed a model whose working is based on the total processing time of the individual jobs. The priority of the jobs in the schedule is done by the concept that the job with the highest in process time has the greatest lead over the others. He utilized this concept and suggested a heuristics for minimizing make span, which is commonly known as NEH heuristics algorithm.

Nagar et al. [7] developed a distinct method for the minimization of mean flow time along with the minimization of make span in a flow shop environment. They used two very different methods to reach to the required objective. These two methods are branch and bound method and the evolutionary algorithm i.e. genetic algorithm. Nowicki and Smutnicki [8] implemented tabu search to solve the flow-shop scheduling problem. Neppalli et al. [9] used the basic evolutionary algorithm for solving the two machine problem. It used the genetic algorithms approach to carry out the same, which aimed at minimizing the make span.

Reodecha et al. [10] evaluated the sequencing of heuristics for flexible flow shop scheduling problems. Biskup and Herrmann [11] together have generated a model which uses due dates as its constraints. This method is applicable only to single machine problems which are past sequence dependent models. They created this model whose sole motive is to minimize the penalties which are incurred if the demand is not fulfilled within the due date.

He and Hui [12] used the concept of evolutionary algorithm for the scheduling of a single-stage and multi-product batch plants along with parallel units. The evolutionary algorithm here used is the genetic algorithm. They used a large sized problem which is first solved using genetic algorithm and then finally it was proposed based on the heuristics approach.

Eren and Guner [13] created a model which involves two different criteria to be satisfied which involves minimizing the total completion time and the other is to minimize the make span. The objective of this process involved the technique of learning effect. Here they considered a two machine flow shop problem of scheduling where the concept of learning is applied to find out the best possible schedule for the minimization of weighted make span.

Tseng and Liao [14] took a m-machine and n-job flow shop scheduling problem whose only to minimize the total earliness and tardiness. Here they have used particle swarm optimization technique as the evolutionary algorithm. Through the swarm technique, they have been successful in finding out the weighted earliness and weighted tardiness which they finally minimized to fulfill the demand of the company.

Wu and Zhou [15] did a stochastic scheduling approach to get the required schedule of jobs. It uses a stochastic environment along with a single machine to carry out the target of minimizing the lateness for the completion of the job. This included a group of jobs with different due dates assigned to them, where we need to find the lateness value of each of the jobs and finally find out the maximum lateness value and then minimize that value.

Mosheiov and Sarig [16] did this work based on the motivation of minimizing the cost factors which indirectly affects the company's revenue. The various cost factors include earliness, lateness, tardiness, number of tardy jobs and the latest due date demanded. This

model comprises of the target to schedule a sequence of jobs that would lead to the minimization of due dates, lateness, tardiness, and other cost factors.

Cheng and Lin [17] did a mixture of various methods including Johnson's rules, the concept of relocation and the introduction of composite jobs. Similar to Johnson's algorithm this method uses the problem of two machines for solving the flow shop scheduling problem. Artificially jobs are made with the same idle time as that of the actual working machine and this helps in minimizing the make span of the flow shop problem. This concept is called that idea of composite jobs. And it aims at minimizing the make span.

Modrak et al. [18] did the comparison between the various heuristics algorithms based on the output value of the make span. The heuristics used are namely Palmer's algorithm, Gupta's algorithm, CDS algorithm, NEH algorithm and MOD heuristics algorithm. The NEH used maximum iterations among all others while palmer only uses one iteration to reach to the definite result. Though palmer's algorithm was very fast but it lacked accuracy and the results obtained from palmer's algorithm don't coincide with the best result among all other heuristics algorithm.

Chia and Lee [19] acquired the concept of learning effect for the development of a model to minimize the completion time. The total completion time in various flow shop environments is minimized as required by this method by introducing the concept of learning. The process speed can also be accelerated by involving various dominance rules and other parameters Lia et.al [20] found a method for achieving the optimal solution by minimization of the total flow time. This method mainly involved composite heuristics models that were put into action while solving for total flow time in permutation flow shop environment. Here the jobs are to be processed under a particular sequence and the heuristics approach developed has worked very well in minimizing the total flow time.

CHAPTER 3: METHODOLOGY

3.1 INTRODUCTION:-

The job shop and flow shop scheduling problems are evaluated for increasing the efficiency of the machines and to obtain the optimal processing data for the same. Various methods has been taken into consideration for this work. These methods involve Johnson's algorithm and Kusiak's algorithm for carrying out the optimal sequences. Then we used branch and bound methods and Shifting bottleneck methods to evaluate the optimal make span in a job shop problem. Even we have carried out many heuristics algorithms to get the best optimal result in a flow shop environment for the efficient working of the machine.

3.2 Johnson's Rule:

This rule is one of the simplest methods to find out the minimum make span. This rule is normally employed to two machines problems. In case of three or more machines, this rule is applied only when either of the following conditions is satisfied. Those two rules are:

- a) Least in process time for machine 1 must be higher than or same as the maximum in process time for machine 2.
- b) Least in process time for machine 3 must be higher than or same as the maximum in process time for machine 2.

Algorithm for Johnson's Method:

1. First of all, the different processing times for every different jobs are assigned to both machines.
2. The least/smallest processing times of each job from all the jobs is noticed and if the minimum processing time of the particular job is on 1st machine , then the job

is put into the first sequence otherwise if the minimum processing time is on 2nd machine , then it is put into the last place of the sequence.

3. The selected job is then removed from the sequence.
4. Then again the minimum processing time of any job from the list of jobs is detected and if the least processing time is on machine 1, and then it is put accordingly in the second position if the previous job is in first position or in the first position if the previous job is in last position. While if the least processing time is on machine 2, then it is put accordingly in the last position if the previous job is in the first position otherwise in the second last position.
5. Accordingly all the jobs are sequenced in the above mentioned procedure.

Step By step Procedure:-

1. Assume $A = \{ j, t_{i1} < t_{i2} \}$ and $B = \{ j, t_{i1} \geq t_{i2} \}$
2. Rank the jobs in A by increasing order of t_{i1}
3. Rank the jobs in B by decreasing order of t_{i2}
4. The optimized order is given by the ordered set $\{A\}$ followed by the ordered set $\{B\}$.

Let's take a 5 job and 2 machine flow shop scheduling problem [22]:

Job i	1	2	3	4	5
$t_{i,1}$	3	5	1	6	7
$t_{i,2}$	6	2	2	6	5

The solution is modelled below:

1- Job sets $A = \{ 1, 3, \}$ and $B = \{ 2, 4, 5 \}$

2- Sort jobs in A as follows:

Job i :	3	1
$t_{i1} :$	1	3

3- Sort jobs in B as follows:

Job i :	4	5	2
t_{i2} :	6	5	2

4- The optimal order is {3, 1, 4, 5, 2}

3.3 Kusiak's Algorithm:

Step by step procedure for sequencing the jobs:

1. Let $p = 1, k = m$
2. For every single operation, strictly note the operation number, smallest in process time and notation of its respective m/c.
3. Rank the list obtained above in non-decreasing order of in process time.
4. For every single listing in the above list, if m/c notation is 1, then
 - a) update the respective operation number at position p ,
 - b) update $p = p + k$
 Otherwise
 - a) update the respective operation number at position p ,
 - b) update $k = k - 1$
5. Stop the process if the whole list gets completed.

Let's take a 5 job and 2 machine flow shop scheduling problem:

Job i	1	2	3	4	5
$t_{i,1}$	3	5	1	6	7
$t_{i,2}$	6	2	2	6	5

3.4 Branch & Bound Method:-

Consider a single machine job shop scheduling with the following values [21]:

p_j = Processing time for job j .

r_j = Earliness for job j .

d_j = Due date for job j .

Job	1	2	3	4
p_i	4	2	6	5
r_i	0	1	3	5
d_i	8	12	11	10

Step 1: - Job 1 is fixed first, then sequencing is done by Preemptive EDD (Earliest Due Date)

Job:-	1	4	3	2
CT	4	10	15	17
DD	8	10	11	12

So, Lower Bound, $LB = 5$

Step 2: - Job 2 is fixed first, then sequencing is done by EDD (Earliest Due Date)

Job:-	2	1	4	3
CT	3	7	12	18
DD	12	8	10	11

So, Lower Bound, $LB = 7$

Step 3: - If job 3 is taken, its arrival is 3 but within that time we could complete job 2 completely. So, without computing we can say $LB(\text{Job } 3) > LB(\text{Job } 2)$.

Similarly, For Job 4, $LB(\text{Job } 4) > LB(\text{Job } 1)$ and $LB(\text{Job } 2)$

So, we need not calculate for job 3 and job 4.

Step-4:-

Since Job 1 is fixed in first place, we will look for 2nd Place.

For 1, 2, __, __

For 1, 3, __, __

Job:-	1	2	4	3
CT	4	6	11	16
DD	8	12	10	11

So, Lower Bound, $LB = 6$

Job:-	1	3	4	2
CT	4	10	15	17
DD	8	11	10	12

So, Lower Bound, $LB = 5$

For 1, 4, __, __

Job:-	1	4	3	2
CT	4	10	16	18
DD	8	10	11	12

So, Lower Bound, LB=6

Therefore we choose 1, 3, __, __.

Now looking for the 3rd and 4th place:-

For 1,3,2,4

Job:-	1	3	2	4
CT	4	10	12	17
DD	8	11	12	10

So, Lower Bound, LB=7

For 1,3,4,2

Job:-	1	3	4	2
CT	4	10	15	17
DD	8	11	10	12

So, Lower Bound, LB=5

⇒ Minimum Lower Bound, LB=5 (For 1,3,4,2)

⇒ Optimal Sequence is 1,3,4,2.

So, by Branch and Bound algorithm we got an optimal Scheduling sequence.

Fig.2 shows the results of the branch and bound algorithm:

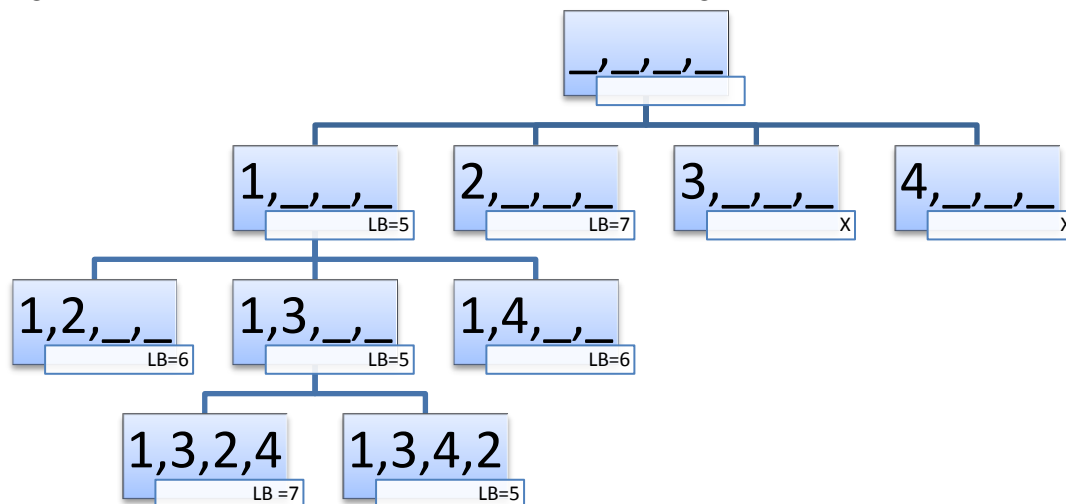


Fig.2: Results of branch and bound algorithm

3.5 Shifting Bottleneck Heuristics:-

Due to some limitations of the branch and Bound algorithm, computing became very complex i.e., we need to evaluate $n!$ Number of times to get to the optimal solution. So, shifting bottleneck heuristics came to rescue.

Considering a 3 job, 3 machine problem with following given data [21]:-

J₁	M₁ (7)	M₃ (8)	M₂ (10)
J₂	M ₂ (6)	M ₁ (4)	M ₃ (12)
J₃	M ₁ (8)	M ₂ (8)	M ₃ (7)

ITERATION-I:-

Total Load on each machine:-

$$M_1 = 7+4+8 = 19$$

$$M_2 = 10+6+8 = 24$$

$$M_3 = 8+12+7 = 27$$

⇒ Make span, $C_{\max} = 27$ (Longest Path)

❖ Start with LB of 27 on M3 because it has the highest load among the 3 machines.

❖ ***M₃ is the bottleneck.***

Applying $1/r_j/L_{\max}$ to Machine-3

LB=27 (M₃)

$d_j = C_{\max} - \text{Node values on right side}$

	J₁	J₂	J₃
P_i	8	12	17
r_i	7	10	16
d_i	17	27	27

❖ First fixing job 1, then job 2 & then job 3. Compare their LB.

For 1, –, –

	1	2	3
CT	15	27	34
DD	17	27	27

LB=7

For 2, –, –

	2	1	3
CT	22	30	37
DD	27	17	27

LB = 13

	3	1	2
CT	23	31	43
DD	27	27	27

For 3, –, –

LB = 24.

Therefore, we choose 1, 2, and 3. Since it has lowest LB. Now since job 1 is fixed we need to locate job 2 and job 3.

For 1, 2, 3

	1	2	3
CT	15	27	34
DD	17	27	27

LB = 7

For 1, 3, 2

	1	3	2
CT	15	23	35
DD	17	27	27

LB = 8

Therefore for Machine 3, the optimal sequence is 1, 2, 3.

Fig.3 shows the result of first iteration of shifting bottleneck heuristics:

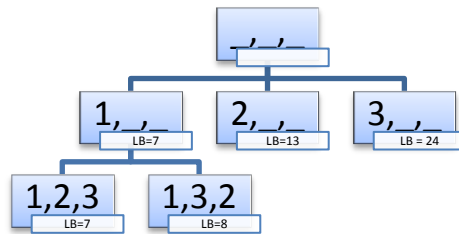


Fig.3: Result of first iteration in shifting bottleneck heuristics

ITERATION-2:-

Now, Longest path = $C_{\max} + \text{Lowest LB} = 27 + 7 = 34$. Therefore New $C_{\max} = 34$.

Total Load on each machine:-

$M_1 = 7+4+8 = 19$; $M_2 = 10+6+8 = 24$

❖ ***M_2 is the next Bottleneck***

Applying $1/r_i/L_{\max}$ to Machine-2

	J₁	J₂	J₃
P_i	10	6	8
r_i	15	0	8
d_i	34	18	27

❖ First fixing job 1, then job 2 & then job 3. Compare their LB.

For 1, __, __

Arrival time of Job 1 is 15, within it we could complete Job 2 completely. So, it is not evaluated.

For 2, __, __

	2	3	1
CT	6	16	26
DD	18	26	34

LB = 0

- ⇒ We got for feasible optimal sequence for Machine 2.
- ⇒ For M/c 2 sequence is 2, 3, 1
- ⇒ So, disjunctive arcs for M_2 is cut out.

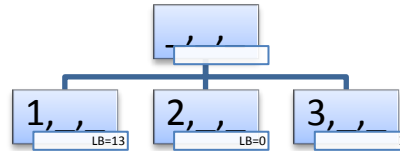


Fig.4: Result of second iteration in shifting bottleneck heuristics

Now, Load on $M_1 = 19$

❖ *So, Machine 1 is the last Bottleneck.*

ITERATION 3:-

Now $C_{\max} = 34$ (Longest Path)

Applying $1/r_j/L_{\max}$

	J₁	J₂	J₃
P_i	7	4	8
r_i	0	6	0
d_i	7	15	16

❖ First fixing job 1, then job 2 & then job 3. Compare their LB.

For 1, _ , _

	1	2	3
CT	7	11	19
DD	7	15	16

LB = 3

For 2, _ , _

	2	1	3
CT	10	17	25
DD	15	7	16

For 3, _ , _

	3	1	2
CT	8	15	19
DD	16	7	15

LB = 8. Therefore, Our Optimal solution has LB = 3.

- ⇒ For Machine 1 the optimal sequence is 1, 2, and 3, Optimal Make span= $34+3=37$

⇒ **Therefore, Optimal Make Span is 37.**

3.6 HEURISTICS ALGORITHM'S:-

1. Palmer's Algorithm.
2. CDS (Campbell Dudek Smith) Algorithm.
3. NEH (Nawaz Ensore Ham) Algorithm.

3.6.1 PALMER'S ALGORITHM:-

This method will try and find out a weighted sum for each of these jobs. So, we will give weights to each of these machines and then we try and find out the weighted sum of each jobs.

Step 1:- Consider a job scheduling problem for m machine and n jobs.

Step 2:- Assign some specific weights to each machine by the formula:-

$$S(j) = - \sum_{t=1}^k [(k - (2 * t - 1)) * O(j, t)]$$

Sep 3:- Evaluate the weight of each job by a given procedure by multiplying the weights with the processing times.

Step 4:- Sort the jobs in the decreasing order of their weights.

Step 5:- Formulate a sequence based on the sorting done in Step 4.

Step 6:- Calculate the Make Span for the above sequence.

Example: - Consider a 3 machine and 5 job flow shop scheduling problem [21].

	M1	M2	M3
J1	16	18	12
J2	14	10	11
J3	13	20	15
J4	19	15	19
J5	15	16	16
Weight	-2	0	+2

Step 1: Assign weights to each machine.

Weight (M1) = -2, Weight (M2) = 0, Weight (M3) = +2

Step 2:- Calculate the weight of each job.

$$W(J1) = (-2*16) + (0*18) + (2*12) = -8; W(J2) = (-2*14) + (0*10) + (2*11) = -6$$

$$W(J3) = (-2*13) + (0*20) + (2*15) = 4; W(J4) = (-2*19) + (0*15) + (2*19) = 0$$

$$W(J5) = (-2*15) + (0*16) + (2*16) = 2$$

Step 3:- Sort in decreasing order

J3-J5-J4-J2-J1

Step 4:- Calculate the make span of the above sequence.

CT	M1	M2	M3
J3	13	33	48
J5	28	49	65
J4	47	64	85
J2	61	74	95
J1	77	95	109

⇒ Make span associated with this sequence is 109.

3.6.2 CDS (Campbell Dudek Smith) Algorithm:-

CDS algorithm uses the Johnson's algorithm at its each iteration so as to reach a better optimal solution.

For m Machine: M1, M2, M3..... M (m) & n Jobs

Step 1:- We create (M-1) Sequence i.e.

S1	M1	M(m)
S2	M1+M2	M(m-1)+M(m)
S3	M1+M2+M3	M(m-2)+M(m-1)+M(m)
....	M1+M2+M3+.....
....	M1+M2+M3+.....
S(m-1)	M1+M2+M3+....+M(m-1)	M2+M3+....+M(m)

Step 2:- Apply Extension of Johnson' Algorithm to each of the above (m-1) sequences.

Step 3:- Take the best possible Make Span out of them.

Step 4:- CDS evaluates (m-1) sequences.

Example: Consider the same problem of 3 machine & 5 Job as in palmer's algorithm.

	M1	M2	M3
J1	16	18	12
J2	14	10	11
J3	13	20	15
J4	19	15	19
J5	15	16	16

Here we have 3 machines so the possible sequences are $(3-1) = 2$, i.e.

S1	M1	M3
S2	M1+M2	M2+M3

Applying Johnson's algorithm to sequence 1 and sequence 2:-

	M1	M3		M1+M2	M2+M3
J1	16	12	J1	34	30
J2	14	11	J2	24	21
J3	13	15	J3	33	35
J4	19	19	J4	24	34
J5	15	16	J5	31	32



J3	J5	J4	J1	J2
----	----	----	----	----

Calculate the make span: -

Calculate the make span:-

CT	M1	M2	M3
J3	13	33	48
J5	28	49	65
J4	47	64	84
J1	63	82	96
J2	77	92	107



J5	J3	J4	J1	J2
----	----	----	----	----

CT	M1	M2	M3
J5	15	31	47
J3	28	51	66
J4	47	66	85
J1	63	84	97
J2	77	94	108

⇒ Make Span is 107.

Now, since the make span of Sequence 1 < Sequence 2

We Choose sequence 1 as the optimal solution.

3.6.3 NEH (Nawaz Ensore Ham) Algorithm:-

This algorithm is known as insertion algorithm. And when this insertion algorithm is used for make span minimization then it is called NEH Algorithm.

Procedure for this NEH Algorithm:-

Step 1:- Calculate the total sum of processing time for each job.

Step 2:- Sort the jobs in the decreasing order of processing times.

Step 3:- Take the first to jobs in the sorted sequence and formulate different combinations.

Step 4:- Calculate the make span for each of the combination.

Step 5:- Select the combination with minimum make span.

Step 6:- Insert the next job from the sequence obtained in Step 2.

Step 7:- Carry out all possible combination of the 3 jobs now.

Step 8:- Repeat Step 4 followed by Step 5 followed by Step 6.

Step 9:- Continue the process till all jobs are completed.

Example: - Consider the same problem of 3 machine & 5 Job as in palmer's algorithm and CDS algorithm.

	M1	M2	M3	Processing Time
J1	16	18	12	46
J2	14	10	11	35
J3	13	20	15	48
J4	19	15	19	53
J5	15	16	16	47

Step 1:- Calculate the processing times as shown above.

Step 2:- Sort in the decreasing order of processing times.

J4	J3	J5	J1	J2
-----------	-----------	-----------	-----------	-----------

Step 3: Take J3 & J4.

Possible combinations: - J3-J4 & J4-J3.

For J3- J4: -

	M1	M2	M3
J3	13	33	48
J4	32	48	67

Therefore we choose J3-J4.

For J4-J3:-

	M1	M2	M3
J4	19	34	53
J3	32	54	69

Step 4:- Then we take the next job in the sequence i.e., J5.

Now J5 can be squeezed in three ways i.e., J5-J3-J4, J3-J5-J4, J3-J4-J5

For J5-J3-J4:-

	M1	M2	M3
J5	15	31	47
J3	28	51	66
J4	47	66	85

For J3-

J5-J4: -

	M1	M2	M3
J3	13	33	48
J5	28	49	65
J4	47	64	84

For J3-J4-J5:-

	M1	M2	M3
J3	13	33	48
J4	32	48	67
J5	47	64	83

So, we choose the sequence J3-J4-J5

Step 5:- Then we take the next job in the sequence i.e., J1.

Now J1 can be squeezed in 4 ways i.e., J1-J3-J4-J5, J3-J1-J4-J5, J3-J4-J1-J5, J3-J4-J5-J1.

For J1-J3-J4-J5:-

	M1	M2	M3
J1	16	34	46
J3	29	54	69
J4	48	69	88
J5	63	85	101

For J3-J1-J4-J5

	M1	M2	M3
J3	13	33	48
J1	29	51	63
J4	48	66	85
J5	63	82	101

For J3-J4-J1-J5:-

	M1	M2	M3
J3	13	33	48
J4	32	48	67
J1	48	66	79
J5	63	82	98

So, we choose sequence J3-J4-J5-J1

For J3-J4-J5-J1

	M1	M2	M3
J3	13	33	48
J4	32	48	67
J5	47	64	83
J1	63	82	95

Step 6:- Then we take the next job in the sequence i.e., J2.

Now J2 can be squeezed in 5 ways

i.e., J2-J3-J4-J5-J1, J3-J2-J4-J5-J1, J3-J4-J2-J5-J1, J3-J4-J5-J2-J1, J3-J4-J5-J1-J2

For J2-J3-J4-J5-J1:-

	M1	M2	M3
J2	14	24	35
J3	27	47	62
J4	46	62	81
J5	61	78	94
J1	77	96	108

For J3-J2-J4-J5-J1

	M1	M2	M3
J3	13	33	48
J2	27	43	59
J4	46	61	80
J5	61	77	96
J1	77	95	108

	M1	M2	M3
J3	13	33	48
J4	32	48	67
J2	46	58	78
J5	61	77	94
J1	77	95	107

For J3-J4-J5-J2-J1

	M1	M2	M3
J3	13	33	48
J4	32	48	67
J5	47	64	83
J2	61	74	94
J1	77	95	107

For J3-J4-J5-J1-J2

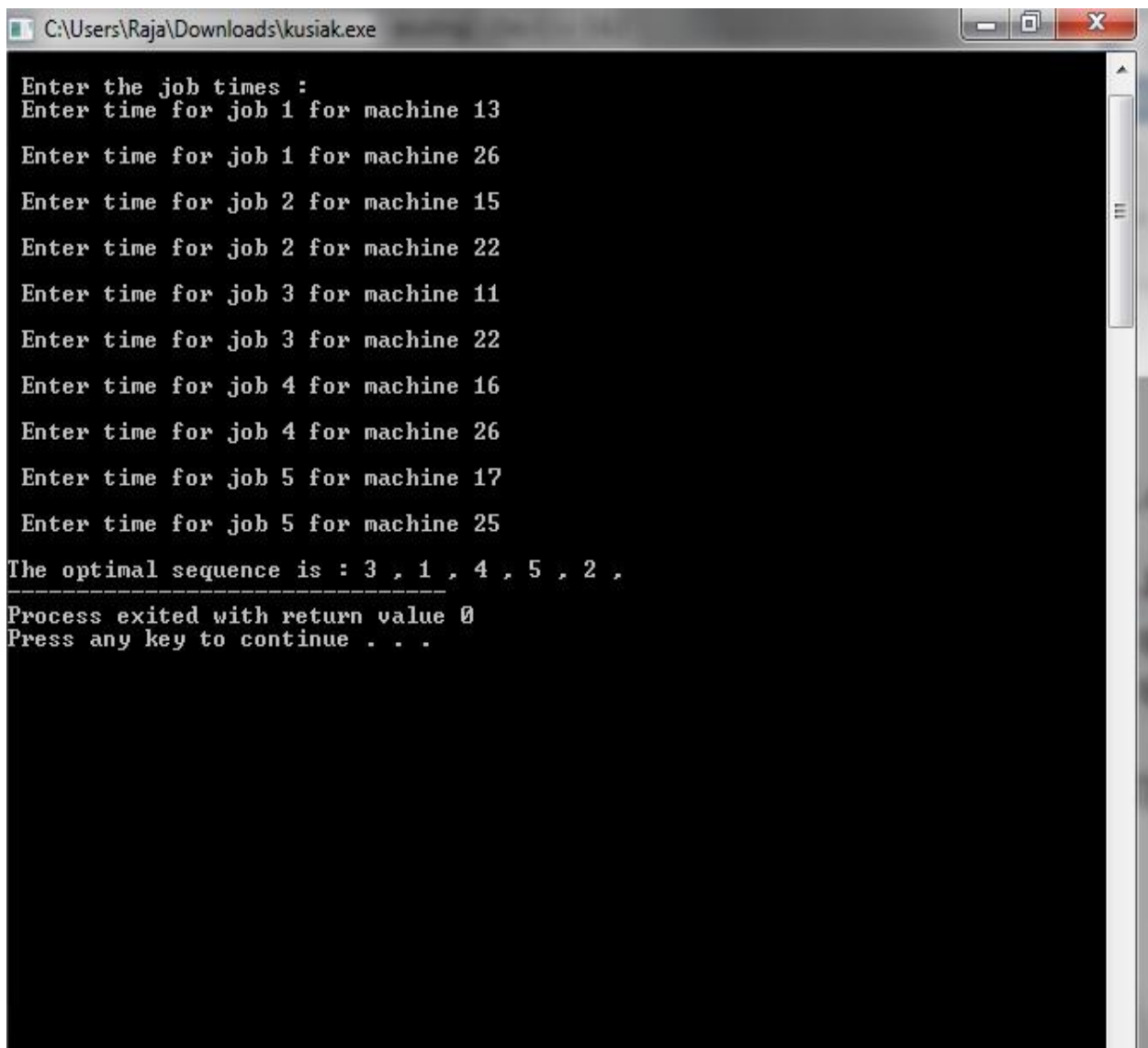
	M1	M2	M3
J3	13	33	48
J4	32	48	67
J5	47	64	83
J1	63	82	95
J2	77	92	106

⇒ Therefore the Optimal Sequence is J3-J4-J5-J1-J2 and optimal make span is 106.

CHAPTER 4: RESULTS

4.1 C Programming Output For Flow Shop Problem Using Kusiak's Algorithm:

Output: Fig.5 gives the output of the flow shop scheduling by Kusiak's algorithm.

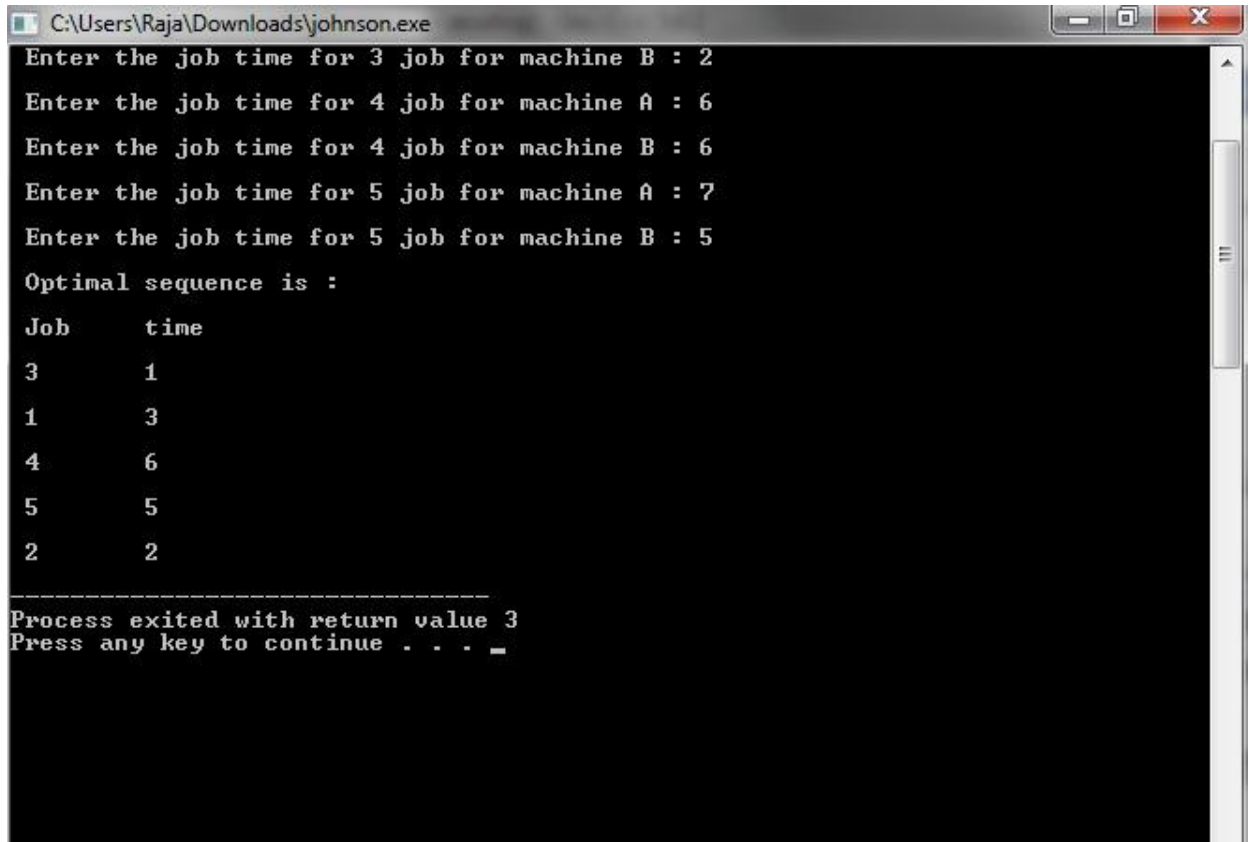
A screenshot of a Windows command prompt window titled "C:\Users\Raja\Downloads\kusiak.exe". The window has a black background with white text. The text shows the program's execution: it prompts for job times for 5 jobs across 5 machines, then displays the optimal sequence as 3, 1, 4, 5, 2, and finally shows the process exit message.

```
C:\Users\Raja\Downloads\kusiak.exe
Enter the job times :
Enter time for job 1 for machine 13
Enter time for job 1 for machine 26
Enter time for job 2 for machine 15
Enter time for job 2 for machine 22
Enter time for job 3 for machine 11
Enter time for job 3 for machine 22
Enter time for job 4 for machine 16
Enter time for job 4 for machine 26
Enter time for job 5 for machine 17
Enter time for job 5 for machine 25
The optimal sequence is : 3 , 1 , 4 , 5 , 2 ,
-----
Process exited with return value 0
Press any key to continue . . .
```

Fig.5: Program output of Kusiak's algorithm

4.2 C Programming Output For Flow Shop Problem Using Johnson's Algorithm:

Output: - Fig.6 gives the output of the flow shop scheduling by Johnson's algorithm.



```
C:\Users\Raja\Downloads\johnson.exe
Enter the job time for 3 job for machine B : 2
Enter the job time for 4 job for machine A : 6
Enter the job time for 4 job for machine B : 6
Enter the job time for 5 job for machine A : 7
Enter the job time for 5 job for machine B : 5
Optimal sequence is :
Job    time
3      1
1      3
4      6
5      5
2      2
-----
Process exited with return value 3
Press any key to continue . . . _
```

Fig.6: Program output of Johnson's algorithm

4.3 Grant Chart for Job shop problem by Shifting Bottleneck Method:

Grant Chart:

Fig.7 gives the Grant chart for shifting bottleneck heuristics:

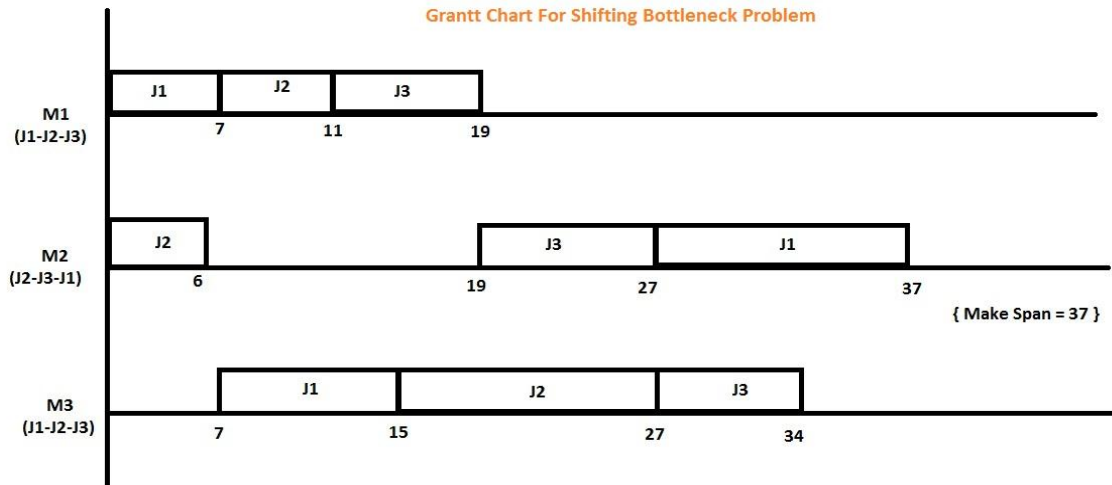


Fig.7: Grant chart for shifting bottleneck heuristics

4.3 Comparison Between the three Heuristics Algorithms:

Comparisons are done between four algorithm namely Johnson's Algorithm, Palmer's Algorithm, CDS Algorithm and NEH Algorithm for minimization of make span in view of following three problems. On comparison it was found that NEH heuristics is the most efficient algorithm among all others.

Problem 1:- 3 Machine, 5 Job Flow shop Problem.

	M1	M2	M3
J1	16	18	12
J2	14	10	11
J3	13	20	15
J4	19	15	19
J5	15	16	16

Optimal Make Span By:-

1. Johnson's Algorithm = 108
2. Palmer's Heuristics = 107
3. CDS Heuristics = 107
4. NEH Heuristics = 106

Problem 2:- 3 Machine, 5 Job Flow shop Problem [17]

	M1	M2	M3
J1	11	12	20
J2	13	15	18
J3	20	10	12
J4	9	12	20
J5	11	18	15

Optimal Solutions By:-

1. Johnson's Algorithm=106
2. Palmer's Heuristics=106
3. CDS Heuristics=106
4. NEH Heuristics=106

Problem 3:- Taillard's benchmark problem: 5 Machine, 20 Job Flow shop Problem [17]

	M1	M2	M3	M4	M5
1	54	79	16	66	58
2	83	3	89	58	56
3	15	11	49	31	20
4	71	99	15	68	85
5	77	56	89	78	53
6	36	70	45	91	35
7	53	99	60	13	53
8	38	60	23	59	41
9	27	5	57	49	69
10	87	56	64	85	13
11	76	3	7	85	86
12	91	61	1	9	72
13	14	73	63	39	8
14	29	75	41	41	49
15	12	47	63	56	47
16	77	14	47	40	87
17	32	21	26	54	58
18	87	86	75	77	18
19	68	5	77	51	68
20	94	77	40	31	28

Optimal Solutions By:-

1. Johnson's Algorithm=1455
2. Palmer's Heuristics=1384
3. CDS Heuristics=1314

CHAPTER 5: CONCLUSIONS

In view of the scheduling work based on the objective of minimization of make span of flow shop and job shop scheduling problems, the following conclusions were carried out:

- Computing by Branch and Bound method was very complex and time consuming, since we need to evaluate $n!$ number of times to get to the optimal solution. So, shifting bottleneck heuristics came as an aid.
- Comparison was done for the best heuristics algorithm for job shop minimization. By our results we found out that for a 5 job 3 machine problem, the make span calculated is 109 for Palmer's algorithm, 107 for CDS algorithm and finally 106 by NEH algorithm. Therefore, from this we concluded that NEH algorithm is more efficient than Palmer's and CDS heuristics.
- For a 'm' machine & 'n' job flow shop problem, number of sequences evaluated by the following:-
 - Palmer's algorithm:- only 1 sequence
 - CDS: - (m-1) Sequences.
 - NEH: - 'n' Sequences.

NEH Heuristics Algorithm will evaluate more sequences than the CDS Algorithm, and also better than CDS for make span minimization.

- Johnson's algorithm for two machine flow shop problem was quite simple one but there is further more scope for optimizing the value of make span obtained by it. Modified Johnson's algorithm for 'm' machines also came handy while carrying out the CDS algorithm.
- Kusiak's algorithm was an excellent method of sequencing but it lacked the capability to find out the optimal make span for the flow shop problem.

REFERENCES

- [1] Pinedo, M. (2010), *Scheduling: Theory, Algorithms, and Systems*, Fourth Edition, Springer, 1-5, DOI 10.1007/978-1-4614-2361-4.
- [2] Johnson, S. M. (1954), Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics*, 1, 61–68.
- [3] Palmer, D. S. (1965), Sequencing jobs through a multi-stage process in the minimum total time – a quick method of obtaining a near optimum, *Operations Research Society*, 16, 101-107.
- [4] Campbell, H. G.; Dudek, R. A. and Smith, M. L. (1970), A Heuristic Algorithm for the n Job, m Machine Sequencing Problem, *Management Science*, 16-10, 630-637.
- [5] Gupta, J. N. D. (1971), A functional heuristic algorithm for the flow shop scheduling problem, 22, 39-47.
- [6] Nawaz, M.; Ensore, E. and Ham, I. (1983), A heuristic algorithm for the m machine, n job flow shop sequence problem, *OMEGA*, 11, 91-95.
- [7] Nagar, A.; Heragu, S. S. and Haddock, J. (1996), A combined branch and bound and genetic algorithm based approach for a flow shop-scheduling problem, *Annals Operation Research*, Springer, 63, 397–414.
- [8] Nowicki, E. and Smutnicki, C. (1996), A fast tabu search algorithm for the permutation flow-shop problem, *European Journal Operation Research*, 91, 160-175.
- [9] Neppalli, V. R.; Chen, C. L. and Gupta, J. N. D. (1996), Genetic algorithms for the two-stage bi criteria flow shop problem. *European Journal of Operational Research*, 95, 356–373.
- [10] Jungwattanakit, J.; Reodech, M.; Chaovalitwongs, P. and Werner, F. (2005), An Evaluation of Sequencing Heuristics for Flexible Flow shop Scheduling Problems. *Annals Operation Research*, Springer, 13, 257–316.

- [11] Biskup, D. and Herrmann, J. (2008), A Single-machine scheduling against due dates with past-sequence-dependent setup times, *European Journal of Operational Research*, 19, 587–592.
- [12] He, Y. and Hui, W. C. (2008), A rule-based genetic algorithm for the scheduling of single-stage multi-product batch plants with parallel units, *Computers and Chemical Engineering*, 32, 3067–3083.
- [13] Tamer, E. and Guner, E. (2008), A bicriteria flow shop scheduling with a learning effect, *Applied Mathematical Modelling*, 32, 1719–1733.
- [14] Tseng, C. and Liao, C. J. (2008), A discrete particle swarm optimization for lot-streaming flow shop scheduling problem, *European Journal of Operational Research*, 191, 360–373.
- [15] Xianyi, W. and Zhou, X. (2008), Stochastic scheduling to minimize expected maximum lateness, *European Journal of Operational Research*, 190, 103–115.
- [16] Mosheiov, G. and Sarig, A. (2009), The due-date assignment on uniform machines, *European Journal of Operational Research*, 193, 49–58.
- [17] Cheng, T.C.E. and Lin, B. (2009), Johnson’s rule, composite jobs and the relocation problem, *European Journal of Operational Research*, 192, 1008–1013.
- [18] Modrak, V.; Semanco, P. and Kulpa, W. (2009), Performance measurement of selected heuristics algorithm for solving scheduling problems *European Journal of Operational Research*, 34, 158–183.
- [19] Wu, C. and Lee, W. C. (2009), A note on the total completion time problem in a permutation flow shop with a learning effect, *European Journal of Operational Research*, 192, 343–347.
- [20] Li, X.; Wang, Q. and Cheng, W. (2009), Efficient composite heuristics for total flow time minimization in permutation flow shops, *Omega*, 37, 155 – 164.
- [21] Srinivasan, G. (2004), Operations and supply management, *Department of management studies*, 7, 27- 29.
- [22] Baker, K. and Trietsch, D. (2009), Principles of sequencing and scheduling, John Wiley and Sons, 230-232.

